Extraction of text-related features for condensing image documents

Dan S. Bloomberg and Francine R. Chen

Xerox Palo Alto Research Center
Palo Alto, CA 94304

ABSTRACT

A system has been built that selects excerpts from a scanned document for presentation as a summary, without using character recognition. The method relies on the idea that the most significant sentences in a document contain words that are both specific to the document and have a relatively high frequency of occurrence within it. Accordingly, and entirely within the image domain, each page image is deskewed and the text regions of are found and extracted as a set of textblocks. Blocks with font size near the median for the document are selected and then placed in reading order. The textlines and words are segmented, and the words are placed into equivalence classes of similar shape. The sentences are identified by finding baselines for each line of text and analyzing the size and location of the connected components relative to the baseline. Scores can then be given to each word, depending on its shape and frequency of occurrence, and to each sentence, depending on the scores for the words in the sentence. Other salient features, such as textblocks that have a large font or are likely to contain an abstract, can also be used to select image parts that are likely to be thematically relevant. The method has been applied to a variety of documents, including articles scanned from magazines and technical journals.

Keywords: image analysis, image segmentation, page segmentation, classification, document image summarization, image morphology, logical analysis, information retrieval

1 Introduction

A summary is commonly thought of as a concise interpretation of a document, represented by a small number of sentences. However, a document summary can take other forms. These may be, in various combinations: extracts, such as sentences or phrases; headings, up to a table of contents; figures; or thumbnails. Taken together, these components can provide different "summary" views of the document.

Techniques for creating computer-generated summaries of textual documents have been developed by a number of researchers (an overview is given in Paice[11]). Because of the difficulty inherent in natural language understanding and generation, most of these techniques rely on extraction, rather than composition, of phrases and sentences.

Although many documents are available as ascii text, many are available only as paper documents. These paper documents can be automatically summarized using techniques developed for text, if the document is first converted

Figure 1: *Text image summarization system.*

to text using optical character recognition (OCR). However, the speed and accuracy of OCR are critically dependent on the text image quality, and in all cases require far more processing time than the summarization process.

In contrast to the text-based techniques developed by other researchers,[11,6,7] this paper describes the image analysis required to create a summary from an imaged document without using OCR. For applications where a fast summary is required, OCR may be both impractical and unnecessary. Such applications may include those in which image data is stored digitally, such as fax transmissions or digital copiers. In these cases, a summary sheet that can be used for later retrieval of the documents may be desired.

Our approach to summarization is an amalgam of document image analysis and methods of text summarization. In the former, it uses a relatively simple generic approach to segmentation, combined with novel methods for extracting higher-level logical elements such as sentences. The primary emphasis in this paper is on the image analysis required to compose summaries. In the latter, it follows work by others on ascii text, in that it does not rely on language understanding or generation, and is described in more detail elsewhere.[4] There are significant differences between text-based methods and ours. Unlike a system that uses OCR, we do not know the characters composing the words. Instead , we classify words by shape into *word equivalence classes*, that are only labelled by an index. In selecting excerpts from text, it is important to ignore *stop words*, which are common words that are not content specific. In text-based systems, these are eliminated by comparing character sequences with pre-defined lists of stop words. By contrast, we must use other information to identify and eliminate stop words. Both approaches have their own errors: incorrect word identity with OCR; incorrect class assignment using word shape matching. An advantage with image-based systems is the fact that scanned images contain useful information not available to text-based systems, such as font size, placement of text, and embedded images.

The overview of our system is shown in Figure 1. The goal is to extract a few short passages, or excerpts, from an imaged document, based on image information and without using ascii representations of words, for presentation as a "summary" of the document. Summaries are presently composed from a small number of selected sentences. To identify a set of summarizing excerpts, the image is segmented to find the text in a galley format, and the location of each word. Word equivalence classes and higher level logical constructs, such as sentences, are generated. Each equivalence class has an image representative and a set of instances belonging to the class. Algorithms based on a statistical characterizations of words in a document, without regard to possible meanings, are then used to determine the keywords, and, from these, the salient sentences for extraction. Finally, the excerpts can be composed to form one or more summary images.

It is necessary to perform both geometric layout analysis and logical layout analysis, to extract *layout* and *logical* information from the page images. These image-based operations are performed in the first three blocks in Figure 1. Layout information describes the manner in which specific components of the document, such as blocks of text and individual words, are spatially organized within the image. Logical information assigns labels or tags to components

of the document.* The logical information that is most important for summarizing the document includes identification of text regions, tagging the text regions based on the relative size of the predominant font within the text region compared to that of the entire document, determining the reading order of selected text regions, finding equivalence classes for all words in selected regions of the document, and locating the sentence and paragraph boundaries.

1.1 Plan of the paper

The organization of the paper is as follows. We first describe, in Section 2, the image analysis that is required to extract text layout information. The process requires a number of steps. The image is deskewed (Section 2.1) and the method for segmenting textblocks is given in Section 2.2. Then in Section 2.3 the font size is determined for the predominant font in the document, and textblocks are sieved for this font size. In Section 2.4, these textblocks are then put into reading order, and in Section 2.5 the textblocks are segmented into textlines and words. Then in Section 3 the word shapes are used to place each word into an equivalence class, using an unsupervised classification process. The detection of sentence boundaries and paragraphs is described in Section 4. At this point, all the information has been extracted directly from the image, and it is used to construct the summary in Section 5. First, in Section 5.1 the stop words are identified and removed from the list of possible keywords. In Section 5.2, criteria for selection of keywords and thematic summary sentences is given, and an example sentence summary is shown. A discussion of some open issues is given in the concluding Section 6.

2 Segmentation

O'Gorman and Kasturi give a summary of existing approaches to page segmentation.[10] Top-down methods typically use either projection profiles of ON and OFF pixels or horizontal closings followed by connected component analysis, to identify textblocks and textlines.[13,12] Region identification can be aided by statistical analysis of underlying textural components, such as runlength histograms. Many of these approaches fail with noisy images, or with complicated text layout that is not exclusively composed of rectangular regions. Bottom-up methods identify groupings, either of pixels by searching for nearest neighbors in different directions,[9] or of connected components, for which statistical information on size can be analyzed. Errors made early in aggregation, where two components were joined that should not have been, can be very difficult to find in later stages.

The outline of our segmentation phase is given in Fig 2. The method is a top-down/bottom-up hybrid, but the essence is top-down because identification of components proceeds from coarse to fine: textblocks, then textlines, and finally individual words. However, each stage proceeds in a bottom-up fashion, where selected pixels are joined to produce the desired image components. The bottom-up segmentation process is at many points guided by size information extracted from the image. We make very few *a priori* assumptions, that are noted as encountered, about the type of document or about its contents. The layout can be complex, with text, images and graphics composed almost arbitrarily. The document images are assumed to be binary.

*There is some overlap between these two. For example, we can think of a word as having both a logical tag and a layout location (or two locations if split across a line break).

Figure 2: *Segmenation*

2.1 Initial image processing: orientation and skew

We assume that the text in each page of the document has a single predominant orientation, and determine that orientation using methods described previously.[2] If no significant orientation is found, the image probably has very little text and can either be analyzed as is, or skipped. After orientation, it is important to remove skew in the image for three reasons: (1) to simplify the segmentation analysis, (2) to improve baseline finding, and (3) to improve the word equivalence classes. The skew angle is determined to within about 0.1 degree,[2] and the image is rotated using two or three orthogonal shears to remove the skew.

2.2 Textblock segmentation

The first step in identifying textblocks is to remove all halftones and other "image" parts. The chosen method has been previously described,[1] and consists of three steps: the formation of a seed image containing pixels exclusively from halftone parts; the formation of a mask image covering all image pixels and with sufficient connectivity to join any halftone seed with the other pixels covering that halftone region; binary reconstruction (filling) from the seed into the mask, resulting in a halftone "mask". This mask is then used to remove the "image" parts, leaving text and line-art.

Next, we identify the textblocks, taking care not to join textblocks in adjacent text columns. This can be done at a resolution of about 75 pixels/inch (ppi). We first make a mask of the vertical whitespace, by inverting the image and opening with a large vertical structuring element. The textblocks are closed, with moderate sized horizontal and vertical structuring elements, to form a single connected component from each textblock. The whitespace mask is then subtracted from the result to insure that adjacent text columns are separated, resulting in a textblock mask.

At this stage in processing, some of the connected components in the textblock mask do not correspond to actual textblocks. For example, various line art components will survive, and must be removed. These components are identified in two ways. Some components, such as horizontal rules, will have a very small height. Others, that can have originated with more elaborate line graphics, can be identified as non-text because they do not have the internal textline structure characteristically found within textblocks. To identify valid textblocks, for each textblock, use a horizontal morphological closing to join the characters in the underlying image (solidifying any textlines that may exist), and analyze the statistics of the resulting "textline" components. The key scaling factor is the median (or, alternatively, the mean) width and height of the resulting "textlines". If the width-to-height ratio is sufficiently large, and if the mean textline width is a significant fraction of the region width, then the region is probably a textblock.

These steps are shown pictorially in Figure 3, starting with the scanned image and ending with identified textblocks. Not shown are the textblock masks corresponding to the final sieving.

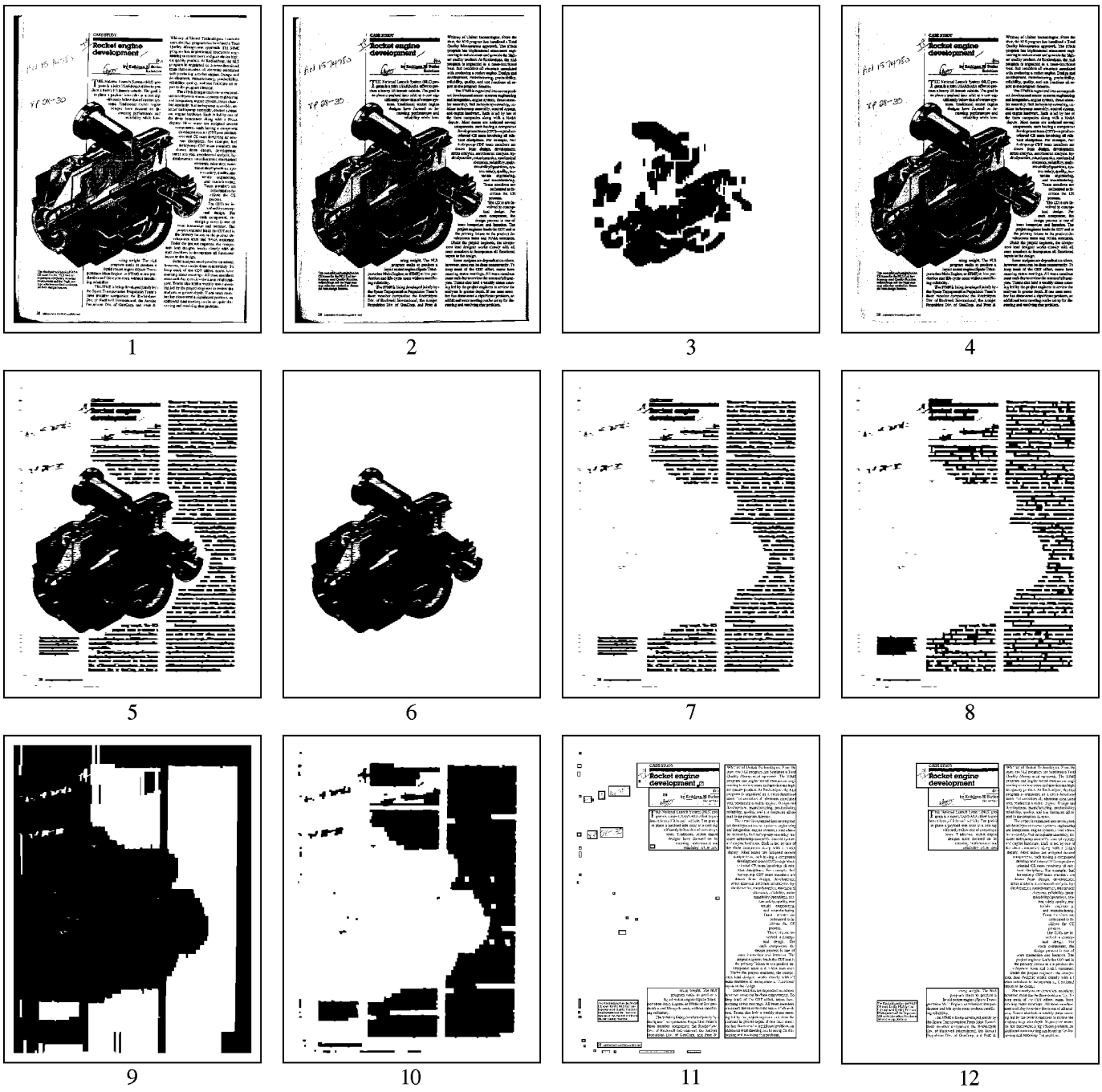


Figure 3: *Early segmentation. The input image (1) is deskewed (2); a mask (4) is thickened to (5), then filled from the seed (3) to form the halftone mask (6). This is subtracted from (5), leaving (7), which is slightly dilated (8). The vertical whitespace mask (9) is constructed and subtracted from a considerably dilated version of (8), leaving a proto-textblock mask (10). These regions (11) are then sieved to remove noise and graphics (12).*

2.3 Dominant font size

Our goal is to identify regions, in reading order, of all textblocks that constitute the main body of text in a document, because keywords, key phrases and sentences will be extracted from this subset of textblocks. The main body of text is usually printed in the *same* font, whereas headings and captions may appear in a variety of fonts. There are two reasons why it is important to identify the main body of text in a document. First, other text, such as headings and captions, is often printed in a different font. As a result, keywords in non-dominant fonts will not match those in the main body, resulting in additional and useless word equivalence classes. Second, and of greater importance, the presence of blocks of non-dominant text, interspersed on the page with the main font, will cause errors in reading order determination, and hence errors in the identification of sentences.

To minimize these errors, textblocks are classified into two sets: those with text whose font size is close to the median size in the document, and those composed of a significantly larger or smaller font. The median textline height for each textblock was previously found. These medians are used to find the median for the entire document. Textblocks with a median height that differs from the median for the document by more than about 12 percent are deemed to be non-conforming and are segregated. Headers and other important information are typically found in textblocks with large textline height.

2.4 Reading order

The conforming textblocks are next analyzed for reading order. Because of inherent layout ambiguities, there is no known method, based only on layout position, that will always give the correct reading order. The general difficulty is that top-to-bottom and left-to-right compete for priority in a complicated and non-standardized way. Nevertheless, the approach described here gives good results in most cases. We model the competition both by using a hierarchical top-to-bottom decomposition and by distinguishing between regions that have either horizontal, vertical or no overlap.

The top-to-bottom decomposition is performed by determining the sets of textblocks that are overlapping in their vertical coordinates. This can be implemented using a horizontal projection profile for the rectangular bounding boxes of regions that constitute the conforming textblocks. If the profile is considered as a set of vertical runs, the set of textblocks that are associated with each run is easily determined from this profile. These sets are strictly ordered from top to bottom.

The remaining (and more difficult) problem is to determine the reading order of the textblocks within each of these sets. Consider two textblocks within such a set. These blocks are typically non-intersecting. If they intersect, we simply choose the block with the highest upper-left corner (or left-most if at the same height) to be first. Suppose they do not intersect. Then there are three different possible situations:

- One is above the other, with horizontal overlap.
- One is left of the other, with vertical overlap.
- There is neither horizontal nor vertical overlap.

The relative order of any two non-intersecting blocks is found from the above three cases, tested in the order given: with horizontal overlap, the higher block is first; otherwise, with vertical overlap, the block to the left is first;

otherwise, with no overlap, choose the block to the left to be first. Because these ordering rules are not transitive, the ordering of a set of textblocks by pair-wise comparisons will, in general, depend on the sequence of comparisons. Nevertheless, the arrangement of textblocks within the sets is usually very simple, the ordering relations are usually transitive, and consequently, the sequence in which comparisons are made is not important.

2.5 Textline and word segmentation

The textlines are located by operations similar to those for finding the size of the font. Separately in each textblock, and at a resolution of about 150 ppi, use a morphological closing with a horizontal structuring element that is sufficiently large to connect all parts of each textline into a single connected component. The bounding boxes of the connected components are found. Components that do not correspond to textlines can be distinguished by size, and are removed; the remaining bounding boxes correspond to actual textlines.

The words are constructed by splitting the textlines, again by merging pixels from the underlying image. For accuracy, this is done in two steps. A small horizontal closing (a 4-pixel structuring element is best at 150 ppi) will join most of the characters in each word for text between 6 and 18 pt, but leaves some words in larger fonts split. The second step uses the bounding boxes of these connected components. After sorting them horizontally within each textline, most of the remaining split words can be joined using a merging operation on the word bounding boxes. The size of the maximum horizontal gap to be closed is scaled proportional to the height of the textlines. It is advantageous to do the final merge on the bounding boxes because the merging distance between characters is often smaller using bounding boxes than morphologically closing on the bitmap. Even with a bounding box merge, punctuation is sometimes not connected to neighboring words, and these small components are removed from the list of words.

3 Word equivalence classes

In this section we describe a method for identifying word images that correspond to the same word[†] without the use of OCR. All words that are sufficiently similar in shape are placed into an equivalence class. The matching parameters must be neither too strict nor too loose. In the former case, instances of the same word will be placed in different classes, whereas in the latter, different words will placed in the same equivalence class. Fortunately, in the summarization application, considerable latitude is available for the criterion of similarity. This is because (1) there is a wide range of matching parameters over which the identification of word equivalence classes is performed with few errors and (2) system performance is not seriously degraded by a small number of errors. Performance of the classifier is significantly degraded by image skew, even of 0.5 degree. It is important to minimize image skew, because classifier performance on multi-page documents is significantly degraded by skew angles greater than ± 0.3 degree.

Word matching is done using a modification of either the blur hit-miss transform (*BHMT*)[3] or similar transforms related to the Hausdorff distance.[8] For the BHMT, the dilated foreground (FG) of the image must contain the FG of the template, and likewise for the background (BG). For the Hausdorff, the dilated FG of the image must contain the FG of the template and the dilated FG of the template must contain the FG of the image. The modification consists of relaxing the containment constraint to permit some number of pixel outliers. This generalizes the BHMT to the *rank* BHMT. For images that are relatively free of pepper noise, and with some tolerance for pixel outliers, the rank versions of BHMT and Hausdorff matching are essentially equivalent. For words in 6 pt or larger, it is sufficient to

[†] In later parts of this paper, we will refer to the class of images that represent the same word as a *term*.

work at a resolution of 150 ppi, and we dilate both FG and BG with a square 3×3 structuring element (SE), and allow a number of pixel outliers that is a specified fraction, between 2 and 4 percent, of the number of pixels in the word image. Only one alignment, where the upper-left corner of the template and image bounding boxes coincide, is tested.

One difficulty with this approach is that the optimum parameters are different for matching small and large words. Suppose that a fraction of the pixel outliers, relative to the image area, are permitted in order to compensate for image variations and misalignment between images of the same word. If the outlier fraction is made sufficiently large to avoid splitting classes for short words, then there will be enough outlier pixels allowed for long words to allow matching of some words that may be otherwise well-aligned but differ by one or two characters. Thus, we optimally should reduce the fraction of outlier pixels allowed for larger words. Another way to achieve this effect is to perform a second matching, using a larger SE for the blur and a much tighter threshold on the number of outliers. The second match can be used to prevent matching of images of different larger words, with little effect on matching of small words. The two matches must both be valid for a word image to be placed in an existing class. We use the notation (SE1, FR1; SE2, FR2) to describe two simultaneous matching conditions. For example, (3, 0.03; 5, 0.002) requires two matches, with the first using a 3×3 SE and with a maximum of 3 percent outliers, and the second using a 5×5 SE and a maximum of 0.2 percent outliers.

Each word in the document is successively analyzed for a match with the representative of an existing class. If a match is found, it is added to the list of instances for that class; otherwise, a new class is formed with the word image as the representative.

One pass of the unsupervised classifier is typically sufficient. For multiple passes, the full set of words is repeatedly classified against a set of representatives, and new representatives are formed from the instances accumulated in each class. In all iterations, new classes can form if a word matches no existing class, and existing classes can be removed if no instances are found to match the representative. As each word is selected, either the *first* (greedy) match to an existing class representative or the *best* match can be used. The greedy algorithm is more efficient, but the best match gives better results and is preferred because either unsupervised classification procedure is relatively fast.

For efficiency, all class representatives are sorted by size, and matches are attempted to a subset differing in width and height by a small amount (typically 2 pixels for 150 ppi images). Further, to optimize matching speed, word-aligned dilated images of each word and class representative are pre-computed, matches are evaluated between these subimages, and matches are aborted when the accumulated number of misses exceeds the allowable outlier fraction in either direction.[3]

69 the	16 teams	13 that	8 also	6 into	5 de-	4 ment,	4 tip	4 Each
55 and	16 are	12 on	8 CE	6 is	5 assembly,	4 Quality	4 in-	4 result,
44 to	16 were	12 have	8 by	6 some	5 assembly	4 As	4 program	4 performance
42 the	15 one	12 be	8 with	6 These	5 pump	4 member	4 they	
37 of	15 as	11 most	7 uct	6 improve	5 reliability,	4 CPI	4 process.	
35 a	14 is	10 fuel	7 has	5 members	5 tem	4 Taguchi	4 number	
21 team	14 turbopump	10 all	7 used	5 noise	4 suction	4 re-	4 est	
18 design	13 was	10 then	6 This	5 to	4 first	4 designs	4 customer	
18 for	13 in	9 each	6 factors	5 three	4 engine	4 which	4 true	
17 The	13 process	9 design,		5 product			4 Some	

Figure 4: *The representatives for the most frequent words in a three-page document, labelled and ordered by the number of occurrences. The equivalence classes were found at 150 ppi, using a non-greedy search, and with matching parameters (3, 0.04; 5, 0.002).*

Figure 5: *Finding sentences and paragraphs*

Figure 4 gives an example of equivalence class representatives for the most frequent words, where all matches are made for two different sets of (blur, misses) parameters in both directions. Typically, the most common word is "the", which is divided into two classes here, and the word "a" is within the set of the ten most common words.

4 Sentence and paragraph identification

It is useful to generate logical information, such as the location of sentences and paragraphs, for the purpose of constructing a summary. Sentences are found first, and then used, along with other layout information, to locate the paragraphs. The overview is shown in Figure 5.

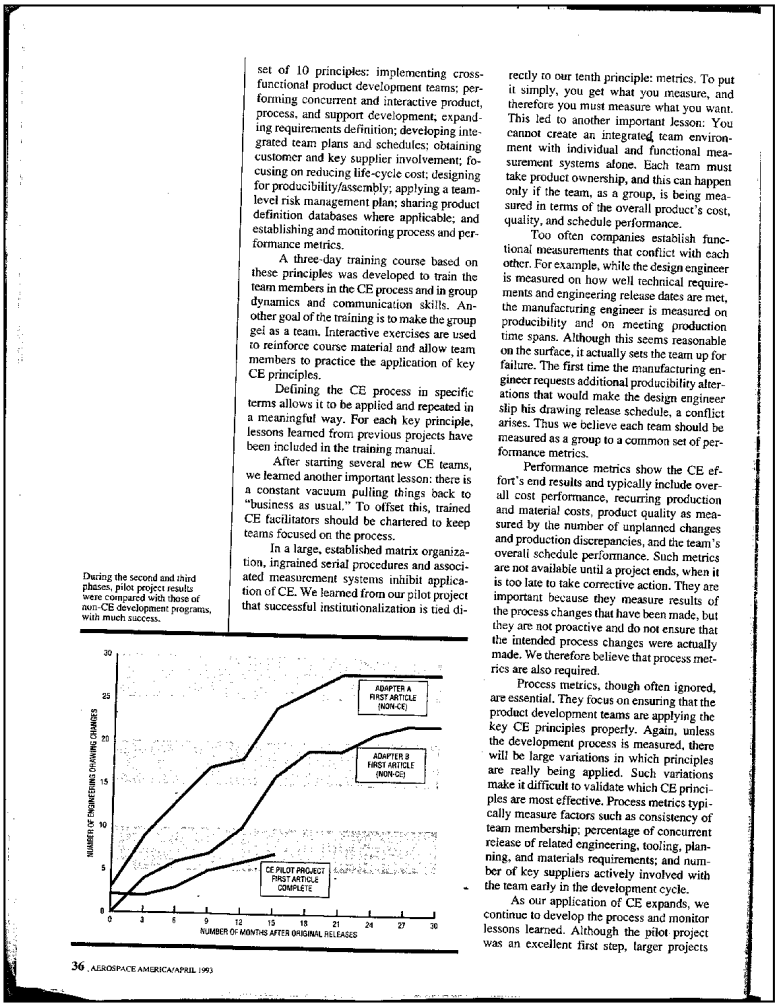
4.1 Sentences

Sentences are identified by searching for periods near the baseline of the textlines, and finding the words most closely associated with the periods. This is considerably slower than the previous steps, because connected component analysis must be done at a resolution of about 300 ppi. Using a 60 MHz Sun Sparcstation 20, sentence labelling on a typical page takes about 2 seconds.

We have previously discussed how to find baselines in deskewed text,[5] using horizontal projection profiles for the image of each textline. Identification of periods is somewhat tricky, because it is necessary to distinguish a period that ends a sentence from noise pixels near the baseline, commas and semicolons, a dot in an ellipsis, the lower dot in a colon, and a dot that ends an intra-sentence abbreviation. We must include periods that are part of question and exclamation marks. And it is necessary to include single and/or double quotes that follow a period and end a sentence as components that are within that sentence.

Most of the following tests require decisions based on measured distances. It is important to use a scale for these comparisons that is based on the size of the font being examined, and is independent of the resolution at which the image is scanned. We choose this scaling parameter to be the measured median height of the bounding boxes of connected components for the characters in the textblock. This is typically the "x-height" of the predominant font.

The procedure we use goes through a sequence of steps, attempting at each to determine one of two different outcomes: (*A*) the dot ends a sentence or (*B*) the dot does not end a sentence. Call these *A* and *B* type decisions respectively. In the following, for clarity, each test is labelled by its decision type. Distances are given as a fraction



set of 10 principles: implementing cross-functional product development teams; performing concurrent and interactive product, process, and support development; expanding requirements definition; developing integrated team plans and schedules; obtaining customer and key supplier involvement; focusing on reducing life-cycle cost; designing for producibility/assembly; applying a team-level risk management plan; sharing product definition databases where applicable; and establishing and monitoring process and performance metrics.

A three-day training course based on these principles was developed to train the team members in the CE process and in group dynamics and communication skills. Another goal of the training is to make the group gel as a team. Interactive exercises are used to reinforce course material and allow team members to practice the application of key CE principles.

Defining the CE process in specific terms allows it to be applied and repeated in a meaningful way. For each key principle, lessons learned from previous projects have been included in the training manual.

After starting several new CE teams, we learned another important lesson: there is a constant vacuum pulling things back to "business as usual." To offset this, trained CE facilitators should be chartered to keep teams focused on the process.

In a large, established matrix organization, ingrained serial procedures and associated measurement systems inhibit application of CE. We learned from our pilot project that successful institutionalization is tied directly to our tenth principle: metrics. To put it simply, you get what you measure, and therefore you must measure what you want. This led to another important lesson: You cannot create an integrated team environment with individual and functional measurement systems alone. Each team must take product ownership, and this can happen only if the team, as a group, is being measured in terms of the overall product's cost, quality, and schedule performance.

Too often companies establish functional measurements that conflict with each other. For example, while the design engineer is measured on how well technical requirements and engineering release dates are met, the manufacturing engineer is measured on producibility and on meeting production time spans. Although this seems reasonable on the surface, it actually sets the team up for failure. The first time the manufacturing engineer requests additional producibility alterations that would make the design engineer slip his drawing release schedule, a conflict arises. Thus we believe each team should be measured as a group to a common set of performance metrics.

Performance metrics show the CE effort's end results and typically include overall cost performance, recurring production and material costs, product quality as measured by the number of unplanned changes and production discrepancies, and the team's overall schedule performance. Such metrics are not available until a project ends, when it is too late to take corrective action. They are important because they measure results of the process changes that have been made, but they are not proactive and do not ensure that the intended process changes were actually made. We therefore believe that process metrics are also required.

Process metrics, though often ignored, are essential. They focus on ensuring that the product development teams are applying the key CE principles properly. Again, unless the development process is measured, there will be large variations in which principles are really being applied. Such variations make it difficult to validate which CE principles are most effective. Process metrics typically measure factors such as consistency of team membership; percentage of concurrent release of related engineering, tooling, planning, and materials requirements; and number of key suppliers actively involved with the team early in the development cycle.

As our application of CE expands, we continue to develop the process and monitor lessons learned. Although the pilot project was an excellent first step, larger projects

During the second and third phases, pilot project results were compared with those of non-CE development programs, with much success.

set of 10 principles: implementing cross-functional product development teams; performing concurrent and interactive product, process, and support development; expanding requirements definition; developing integrated team plans and schedules; obtaining customer and key supplier involvement; focusing on reducing life-cycle cost; designing for producibility/assembly; applying a team-level risk management plan; sharing product definition databases where applicable; and establishing and monitoring process and performance metrics.

A three-day training course based on these principles was developed to train the team members in the CE process and in group dynamics and communication skills. Another goal of the training is to make the group gel as a team. Interactive exercises are used to reinforce course material and allow team members to practice the application of key CE principles.

Defining the CE process in specific terms allows it to be applied and repeated in a meaningful way. For each key principle, lessons learned from previous projects have been included in the training manual.

After starting several new CE teams, we learned another important lesson: there is a constant vacuum pulling things back to "business as usual." To offset this, trained CE facilitators should be chartered to keep teams focused on the process.

In a large, established matrix organization, ingrained serial procedures and associated measurement systems inhibit application of CE. We learned from our pilot project that successful institutionalization is tied directly to our tenth principle: metrics. To put it simply, you get what you measure, and therefore you must measure what you want. This led to another important lesson: You cannot create an integrated team environment with individual and functional measurement systems alone. Each team must take product ownership, and this can happen only if the team, as a group, is being measured in terms of the overall product's cost, quality, and schedule performance.

Too often companies establish functional measurements that conflict with each other. For example, while the design engineer is measured on how well technical requirements and engineering release dates are met, the manufacturing engineer is measured on producibility and on meeting production time spans. Although this seems reasonable on the surface, it actually sets the team up for failure. The first time the manufacturing engineer requests additional producibility alterations that would make the design engineer slip his drawing release schedule, a conflict arises. Thus we believe each team should be measured as a group to a common set of performance metrics.

Performance metrics show the CE effort's end results and typically include overall cost performance, recurring production and material costs, product quality as measured by the number of unplanned changes and production discrepancies, and the team's overall schedule performance. Such metrics are not available until a project ends, when it is too late to take corrective action. They are important because they measure results of the process changes that have been made, but they are not proactive and do not ensure that the intended process changes were actually made. We therefore believe that process metrics are also required.

Process metrics, though often ignored, are essential. They focus on ensuring that the product development teams are applying the key CE principles properly. Again, unless the development process is measured, there will be large variations in which principles are really being applied. Such variations make it difficult to validate which CE principles are most effective. Process metrics typically measure factors such as consistency of team membership; percentage of concurrent release of related engineering, tooling, planning, and materials requirements; and number of key suppliers actively involved with the team early in the development cycle.

As our application of CE expands, we continue to develop the process and monitor lessons learned. Although the pilot project was an excellent first step, larger projects

Figure 6: Example page image (left) and display of some of the analysis (right). The page was deskewed, graphics and text in non-conforming font sizes were removed, the remaining text was segmented down to the word level, and the sentences were found. The first word in each sentence is highlighted.

of the x-height, or in pixels at 300 ppi.

The first step is to order the connected component bounding boxes from left to right within each textline. Connected components with either dimension smaller than 3 pixels are ignored. To past the first tests, the component must be "period-shaped" and within 2 pixels vertically of the computed baseline. The condition for being period shaped is that neither the maximum width nor height exceeds 0.4 (of the x-height) and the difference between the height and width does not exceed 0.12 (of the x-height). *(B): if the "dot" is too small (e.g., pixel noise), or if it doesn't have an allowable shape or proximity to the baseline, ignore it.*

If the dot passes these tests, check for an ellipsis. *(B): if the next component is also a valid dot near the baseline, then it is not a period.*

The dot may be the lower part of a colon. *(B): if the previous or following component is also a dot, and is located directly above the candidate dot at a center-to-center distance that does not exceed the measured x-height, it is part of a colon and not a period.*

The dot may be the lower part of an exclamation or question mark. Check the following component. *(A): if a vertical line through the dot touches this component, and the component is entirely above the dot, then it ends the sentence.*

The final test is the most difficult: to differentiate between an intra-sentence abbreviation and a period. First, check if the left edge of the following component is within 0.4 (of the x-height) of the right edge of the component. *(B): if the following component is too close to the dot, and does not originate from the textline below, then the dot may be part of an intra-sentence abbreviation, and should be ignored.* Otherwise, to determine if the next component starts a new sentence, check its height. If it does not descend below the baseline and extends above the baseline by a distance near the maximum distance for characters on the textline, then it has the shape and location of a capital letter. *(B): if the next component does not have the shape and location of a capital letter, the dot should not end the sentence.* Otherwise, *(A): the dot is assumed to be a period.* This final test will not miss any true periods, but it will mis-classify some intra-sentence abbreviations as periods.

Once the periods have been located, the words that end sentences are tagged as those whose bounding box has a right side closest to each period. All the words in the predominant font have now been put into reading order and labelled by their location in the document, their equivalence class, and the sentence to which they belong. The text analysis can now be performed on these tokens, with no reference back to the image except for image composition for output display.

Figure 6 gives a view of the result of the foregoing processing, on the third page of a scanned document. The original image is on the left. On the right, the textblocks in the predominant font have been segmented and deskewed, and the location of the sentences is indicated by by video-inversion of the first word in each sentence.

4.2 Paragraphs

There are two primary ways in which paragraphs can be laid out: indentation of the first line or extra interline spacing. One approach to locating paragraphs is first to determine the layout method and whether the textlines are left and/or right justified, and then to use this information to decide if each textline, as it appears in sequence, begins a new paragraph. This has a disadvantage in that if extra interline spacing is used, it may not be possible to tell if the first textline of a new textblock begins a new paragraph.

Figure 7: *Finding key sentences.*

We choose another method here, that uses the previously located sentences and ignores the two layout characteristics. It is not sufficient only to locate each sentence whose last word is the right-most word on a textline, because about ten percent of sentences that begin a new textline do not begin a new paragraph. So we use a second criterion: the last word in the sentence must also have a right-hand edge that is significantly to the left of the right-hand edge of the right-most word in the textline above. For right-justified text, this fails when the last word reaches the right margin of the text column, when the textline above is itself a very short paragraph, or when the textline ending the paragraph is the first line in a textblock.

The resulting failure rate, due almost entirely to false negatives, is greater for narrow text columns. These failures can be reduced by checking layout cues (indentation and/or vertical separation) for new paragraphs. Other layout information, such as the location of section headers between textblocks, can also be used, because all sections begin with a new paragraph.

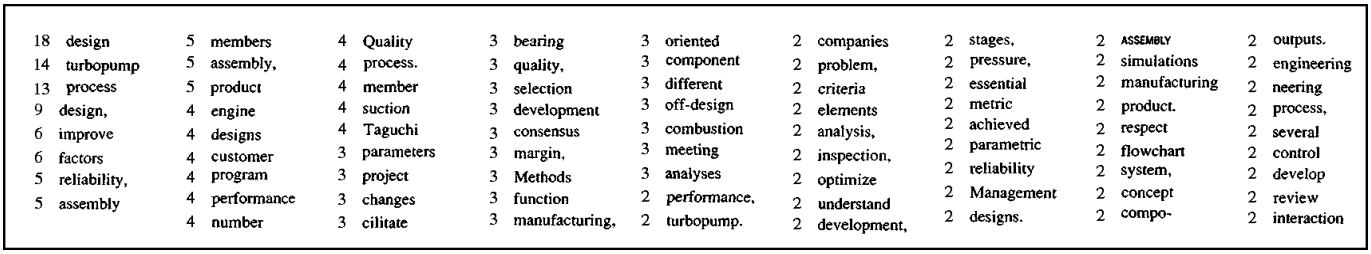
5 Selection of summary sentences

Figure 7 illustrates the steps in identifying key summarizing sentences. To create a summary from an imaged document, information about the word equivalence classes and sentence boundary locations are used to identify stop-words. These words are removed from consideration, and simple statistical measures based on the remaining content words are computed, both over the document and for each sentence in the document. These measures are then used to select key summary sentences.

In text summarization, the characters composing a word are known, so that words in different fonts and cases can be compared. However, in imaged text, character identities are unknown and OCR is required to make similar cross-font comparisons. Our methods are based on high frequency words, without OCR, and we remove text in non-dominant sized fonts from the equivalence classes. Even if these words had been classified, it is unlikely that they would have high enough frequency to be selected.

5.1 Stop-word identification

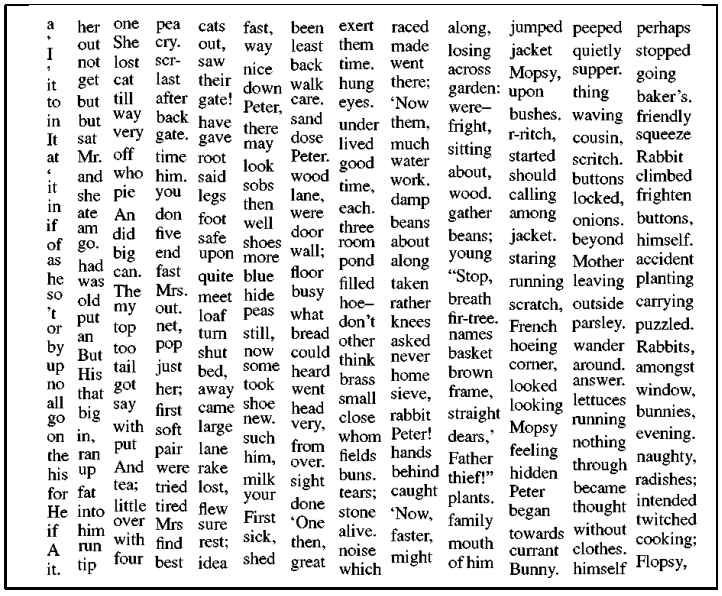
The first step in selecting sentences for presentation as a summary is to eliminate stop words from consideration as terms in the list of word equivalence classes. In text-based systems, a fixed, pre-defined list of stop-words, or *stop-list*, composed of non-content words such as prepositions, articles, adverbs, and letters of the alphabet is often used. For an image-based system that does not perform OCR, the stop-words must be identified based on word shape and the statistics of their occurrence.



18	design	5	members	4	Quality	3	bearing	3	oriented	2	companies	2	stages,	2	ASSEMBLY	2	outputs.
14	turbopump	5	assembly,	4	process.	3	quality,	3	component	2	problem,	2	pressure,	2	simulations	2	engineering
13	process	5	product	4	member	3	selection	3	different	2	criteria	2	essential	2	manufacturing	2	neering
9	design,	4	engine	4	suction	3	development	3	off-design	2	elements	2	metric	2	product.	2	process,
6	improve	4	designs	4	Taguchi	3	consensus	3	combustion	2	analysis,	2	achieved	2	respect	2	several
6	factors	4	customer	3	parameters	3	margin,	3	meeting	2	inspection,	2	parametric	2	flowchart	2	control
5	reliability,	4	program	3	project	3	Methods	3	analyses	2	optimize	2	reliability	2	system,	2	develop
5	assembly	4	performance	3	changes	3	function	2	performance,	2	understand	2	Management	2	concept	2	review
		4	number	3	cilitate	3	manufacturing,	2	turbopump.	2	development,	2	designs.	2	compo-	2	interaction

Figure 8: *The set of content words, ordered by the number of occurrences. The equivalence classes are found as in Fig. 4 and the stop-words are eliminated using word width. The keywords are then chosen as a set of the most frequent words in this list.*

It is evident from Figure 4 that many stop-words are short and of high frequency,[7] whereas content words tend to be longer. Of the content words, the most important for a particular document occur with the greatest frequency. Thus, a simple approach is to remove all short words that may be stop-words, and rank the remaining content words in order of decreasing frequency. Figure 8 shows the equivalence class representatives for the most frequent keywords. This was derived from Fig. 4 by eliminating all representatives shorter than twice the width of the most common one (in this case, "the"). This will not work when the most common word is "a", and this situation is eliminated by requiring that the word width is significantly larger than its height. Although the longer keywords are found, this approach fails to find some shorter content words. For example, in Figure 4 short content words such as "team", "teams", and "fuel" are eliminated as stop-words.



a	her	one	pea	cats	fast,	been	exert	raced	along,	jumped	peeped	perhaps
'	out	She	cry.	out,	way	least	them	made	losing	jacket	quietly	stopped
I	not	lost	scr-	saw	nice	back	time.	went	across	Mopsy,	supper.	going
,	get	cat	last	their	down	walk	hung	there;	garden:	upon	thing	baker's.
it	but	till	after	gate!	Peter,	care.	eyes.	'Now	were–	bushes.	waving	friendly
to	but	way	back	have	there	sand	under	them,	fright,	r-ritch,	cousin,	squeeze
in	sat	very	gate.	gave	may	dose	lived	much	sitting	started	scritch.	Rabbit
It	Mr.	off	time	root	look	Peter.	good	water	about,	should	buttons	climbed
at	and	who	him.	said	sobs	wood	time,	work.	wood.	calling	locked,	frighten
'	she	pie	you	legs	then	lane,	each.	damp	gather	among	onions.	buttons,
it	ate	An	don	foot	well	were	three	beans	beans;	jacket.	beyond	himself.
in	am	did	five	safe	shoes	door	room	about	young	staring	Mother	accident
if	go.	big	end	upon	more	wall;	pond	along	"Stop,	running	leaving	planting
of	had	can.	fast	quite	blue	floor	filled	taken	breath	scratch,	outside	carrying
as	was	The	Mrs.	meet	hide	busy	hoe–	rather	fir-tree.	French	parsley.	puzzled.
he	old	my	out.	loaf	peas	what	don't	knees	names	hoeing	wander	Rabbits,
so	put	top	net,	turn	still,	bread	other	asked	basket	corner,	around.	amongst
't	an	too	pop	shut	now	could	think	never	brown	looked	answer.	window,
or	But	tail	just	bed,	some	heard	brass	home	frame,	looking	lettuces	bunnies,
by	His	got	her;	away	took	went	small	sieve,	straight	Mopsy	running	evening.
up	that	say	first	came	shoe	head	close	rabbit	dears,'	feeling	nothing	naughty,
no	big	with	soft	large	new.	very,	whom	Peter!	Father	hidden	through	radishes;
all	in,	put	pair	lane	such	from	fields	hands	thief!"	Peter	became	intended
go	ran	And	were	rake	him,	over.	buns.	behind	plants.	began	thought	twitched
on	up	tea;	tried	lost,	milk	sight	tears;	caught	family	towards	without	cooking;
the	fat	little	tired	flew	your	done	stone	'Now,	mouth	currant	clothes.	Flopsy,
his	into	over	Mrs	sure	First	'One	alive.	faster,	of him	Bunny.	himself	
for	him	with	find	rest;	sick,	then,	noise	might				
He	run	four	best	idea	shed	great	which					
if	tip											
A												
it.												

Figure 9: *Terms in a sample document that are most likely to be stop-words, ordered by columns with decreasing likelihood.*

Content words tend to occur at the beginning of a sentence, as do some short stop-words. Consequently, when identifying stop-words it is useful to consider if a word is the first content word in a sentence, in addition to its width and frequency. Each term is scored such that longer words appearing at the beginning of a sentence are favored as

content words, as well as words that occur relatively infrequently in the document (see[4] for details). As before, the width of a word image is normalized by the estimated width of "the". The actual width of each word can be used in finding its score because only text in the predominant font is analyzed, and hence it is unnecessary to normalize the width by font size.

A score is computed that ranks each equivalence class (term) by the likelihood that it is a stop-word. Figure 9 shows the terms, ordered by decreasing score as stop-words, for an imaged document about a rabbit named Peter. Note that although the term "Peter" is relatively short, it is in the 11th column of terms, which is mostly composed of longer terms. The earlier method illustrated in Figure 8 would classify such short terms as stop-words because it relies on word length alone. Some instances of "Peter" that occur at the end of a sentence can be observed to occur earlier in the sorted list. The instances fall into different equivalence classes because those at the end of sentences include the period. These sentence-final instances rank higher as stop-words because they are unlikely to be the first content word in a sentence, whereas many of the non-sentence-final instances of "Peter" are not preceded by any content words.

From the list of terms, the first D that are most likely to be stop-words are used as the stop-list in sentence and keyword selection. D is proportional to the length of the document for short documents, because shorter documents are expected to contain smaller subsets of all stop-words. For longer documents, D is set to a value corresponding to the number of possible stop-words, incremented by a factor allowing for the splitting of some terms into more than one equivalence class. It is observed that the value of D is not critical in sentence selection, because a small number of stop-words erroneously included in the term list used to select sentences do not significantly affect most sentence scores.

5.2 Keyword and sentence selection

With the stop-words removed, the K most frequent words are selected from the list of terms as keywords. In the case of a tie, words with the widest bounding boxes are selected, reflecting the fact that wider words are more likely to have more characters, and consequently are more likely to be content words.[‡] These keywords can then be used to select the N highest scoring sentences as summary sentences, where N is optionally specified by the user. K should not be chosen independently of N, because the keywords are used to score sentences based on their "thematic relevance", and the number of themes that can be reasonably covered in the summary is dependent on the number of summary sentences.[4] Thus, K should be chosen as a function of the number of summary sentences N. If K is chosen to be less than N, then the selected key summary sentences cover a small number of themes, resulting in a relatively cogent summary. If K is chosen to be larger than N, then a wider number of topics in the document may be selected, but at the expense of the summary sentences being less coherent as a group.

Using only the keywords, each sentence is assigned a thematic score based on the occurrences of each keyword in the sentence, and weighted by the frequency with which each keyword occurs relative to others in the document. Figure 10 shows the thematic sentences that were selected from the document from which Figure 8 was derived. Five sentences were chosen, and four keywords were used.

[‡]Use of width rather than the number of characters can result in selection of words with fewer characters. This is unavoidable without OCR, because the number of characters can differ from the number inferred from connected components, due to broken or merged characters.

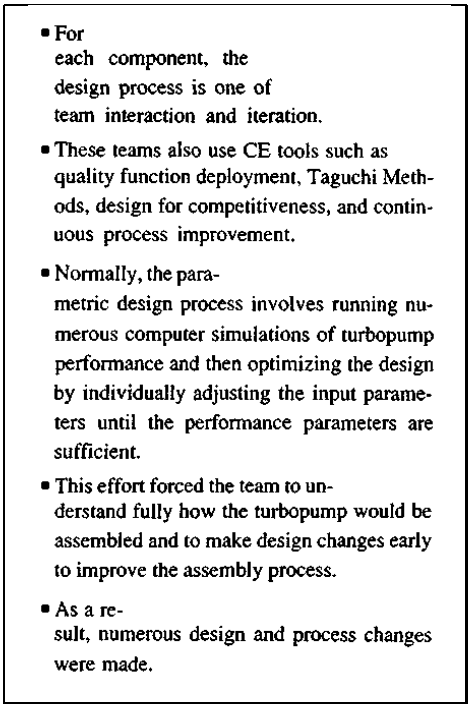


Figure 10: $N = 5$ *selected thematic summary sentences.*

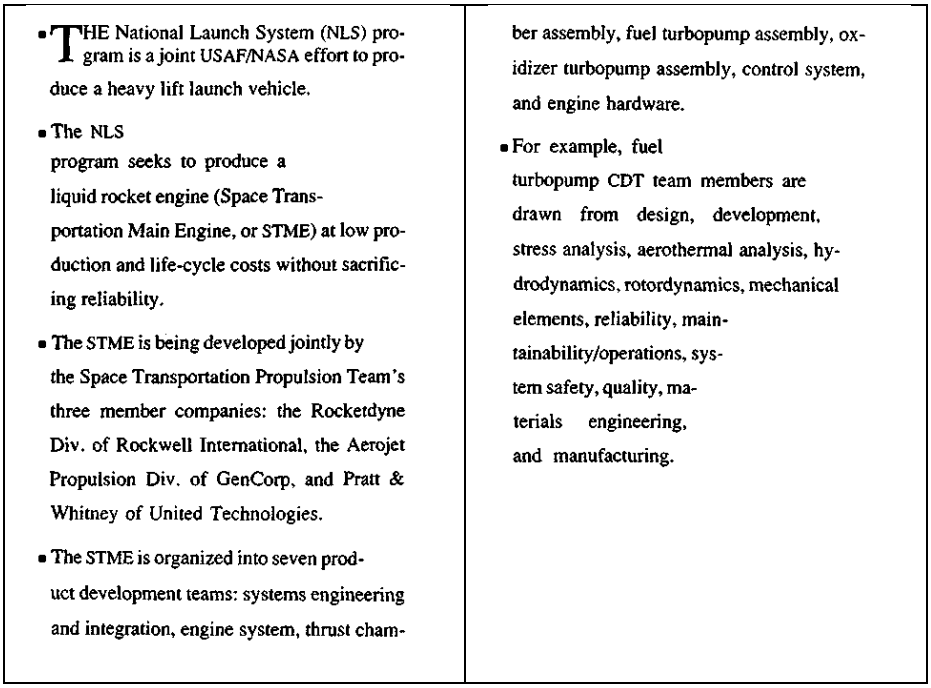
6 Remarks

We have described a system for selecting sentence extracts to serve as a summary of an imaged document. The system does not require the use of OCR. In our image-based system, equivalence classes of words are identified by word image matching, which is far less computationally expensive than OCR. Stop words are identified based on word features that include the frequency, location, and size of words. The processing to identify phrase and sentence extracts is performed only on those text image regions that have been identified as being in the dominant font. Sentences are selected for extraction by identifying a small number of thematic terms and ranking each sentence based on the thematic terms contained in the sentence, and on the location of the sentence within a paragraph and the document as a whole.

The image-based system is about an order of magnitude faster than a system that uses OCR on the full text. However, errors are introduced in the image-based system at several places. In particular, reading order may be incorrect due to either page layout ambiguities or the insertion of like-sized but logically unrelated text, such as text inserts into images, figure captions and footers. These galley errors can cause mutation of sentences.

An advantage in ascii-based systems is that they can more accurately identify stop words. This problem can be mitigated in our system by constructing a hybrid system that performs OCR on the representatives of only those equivalence classes that have sufficient population to qualify as keywords. For a large document, this might involve OCR on several hundred words, which should be an acceptable computational overhead.

Working with text-based systems, Kupiec[6] has shown that it is useful to use the location of a sentence within a



- THE National Launch System (NLS) program is a joint USAF/NASA effort to produce a heavy lift launch vehicle.
- The NLS program seeks to produce a liquid rocket engine (Space Transportation Main Engine, or STME) at low production and life-cycle costs without sacrificing reliability.
- The STME is being developed jointly by the Space Transportation Propulsion Team's three member companies: the Rocketdyne Div. of Rockwell International, the Aerojet Propulsion Div. of GenCorp, and Pratt & Whitney of United Technologies.
- The STME is organized into seven product development teams: systems engineering and integration, engine system, thrust chamber assembly, fuel turbopump assembly, oxidizer turbopump assembly, control system, and engine hardware.
- For example, fuel turbopump CDT team members are drawn from design, development, stress analysis, aerothermal analysis, hydrodynamics, rotordynamics, mechanical elements, reliability, maintainability/operations, system safety, quality, materials engineering, and manufacturing.

Figure 11: $N = 5$ *selected indicative summary sentences, using paragraph information.*

paragraph and the document as a whole when determining the relevance score for that sentence. The selected sentences are referred to as *indicative*, in contrast with the *thematic* sentences found by ignoring the sentence location. We have done preliminary work to generate indicative sentence summaries by combining the thematic sentence information with information about the sentence location. Sentences that occur earlier in a paragraph, or in a paragraph that occurs earlier in the document, are given a greater score for indicative sentences.

An example result is shown in Figure 11, for the same document used to generate the thematic summary sentences shown in Figure 10. We are currently examining ways to produce summaries that are both indicative and thematic.

In addition to identifying a small set of key summarizing sentences, other types of information can be combined to form a summary. Text in font sizes significantly larger than the predominant font are often headings that can be used to create a table of contents. Large text at the beginning can be inferred to be a title. Captions can often be distinguished by a font size different from that of the predominant font and by location. An abstract may be identified based on use of a smaller font size, and/or its location as a prominently featured textblock at the beginning of the document. If found, an abstract can be presented as part of the summary.

A small number of multi-word phrases composed of content words can be identified from the document for presentation as keyphrases, as part of a summary. Like the keyword list, these phrases can indicate the range of topics covered in a document, complementing the summary sentences. Finally, it may be useful to include reduced images of selected figures or pages. For example, a title in 14 pt font on the first page of a document is legible when printed at 3x reduction.

7 Acknowledgements

We are grateful to several current and former members of the PARC community, including Rick Wesel, Dan Huttenlocher, Eric Jaquith, Ramana Rao and John Tukey for their interest and contributions in this research area. We are especially grateful to Meg Withgott for introducing us to the topic. We also thank Jan Pedersen and Kris Halvorsen for their encouragement in this work.

8 REFERENCES

[1] D. S. Bloomberg, "Multiresolution morphological analysis of document images", *SPIE Conf. 1818, Visual Communications and Image Processing '92*, Boston, MA, pp. 648-662, Nov 18-20, 1992.

[2] D. S. Bloomberg, G. E. Kopec and L. Dasari, "Measuring document image skew and orientation", *SPIE Conf. 2422, Document Recognition II*, San Jose, CA, pp. 302-316, Feb 6-7, 1995.

[3] D.S. Bloomberg and L. Vincent, "Blur Hit-Miss transform and its use in document image pattern detection", *SPIE Conf. 2422, Document Recognition II*, San Jose, CA, pp. 278-292, Feb 6-7, 1995.

[4] F.R. Chen and D.S. Bloomberg, "Extraction of Thematically Relevant Text from Images," to appear in Proceedings of the Symposium on Document Analysis and Information Retrieval, Las Vegas, NV, April 1996.

[5] F.R. Chen, D.S. Bloomberg and L. Wilcox, "Spotting phrases in lines of imaged text", *SPIE Conf. 2422, Document Recognition II*, San Jose, CA, pp. 256-269, Feb 6-7, 1995.

[6] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, WA, pp. 68-73, 1995.

[7] H.P. Luhn, "The automatic creation of literature abstracts," *IBM J. Res. Develop.*, vol. 2, pp. 159-165, 1959.

[8] G. Matheron, *Random Sets and Integral Geometry*, J. Wiley and Sons, NY, 1975.

[9] L. O'Gorman, "The document spectrum for page layout analysis", *IEEE Trans PAMI*, Vol. 15, pp. 1162-1173, 1993.

[10] L. O'Gorman and R. Kasturi, *Document Image Analysis*, IEEE Computer Soc. Press, 1995, pp. 165-173.

[11] C. Paice, "Constructing literature abstracts by computer: Techniques and prospects," *Information Processing and Management*, vol. 26, pp. 171-186, 1990.

[12] D. Wang and S. N. Srihari, "Classification of Newspaper Image Blocks Using Texture Analysis", *CVGIP, Vol 47*, pp. 327-352, 1989.

[13] K. Y. Wong, R. G. Casey, and F. M. Wahl, "Document Analysis System", *IBM J. Res. Dev., Vol 26*, pp. 647-656, 1982.